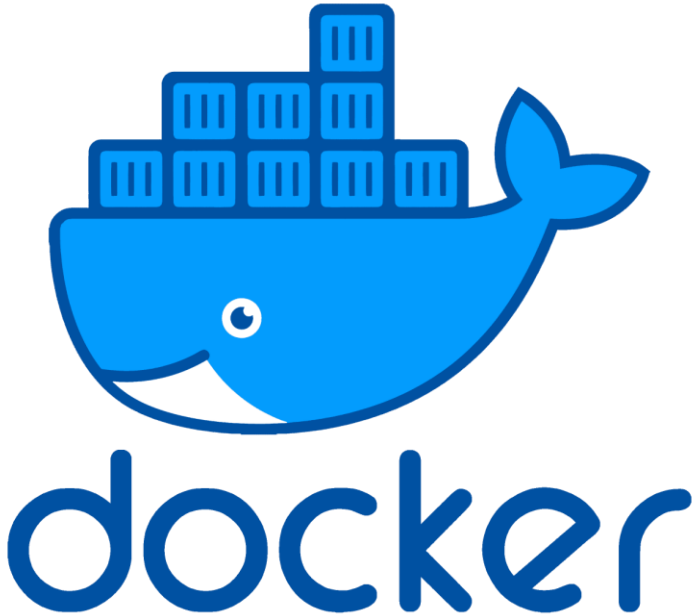
Docker intro training

Mateusz Bączkowski

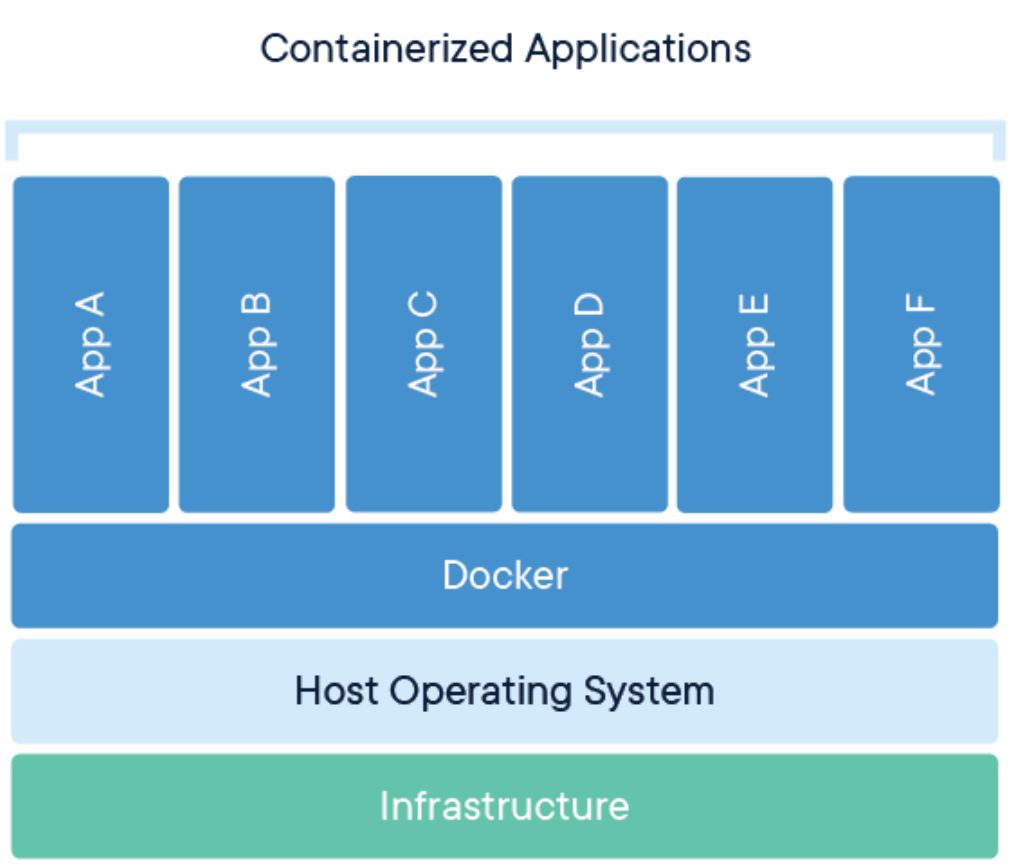


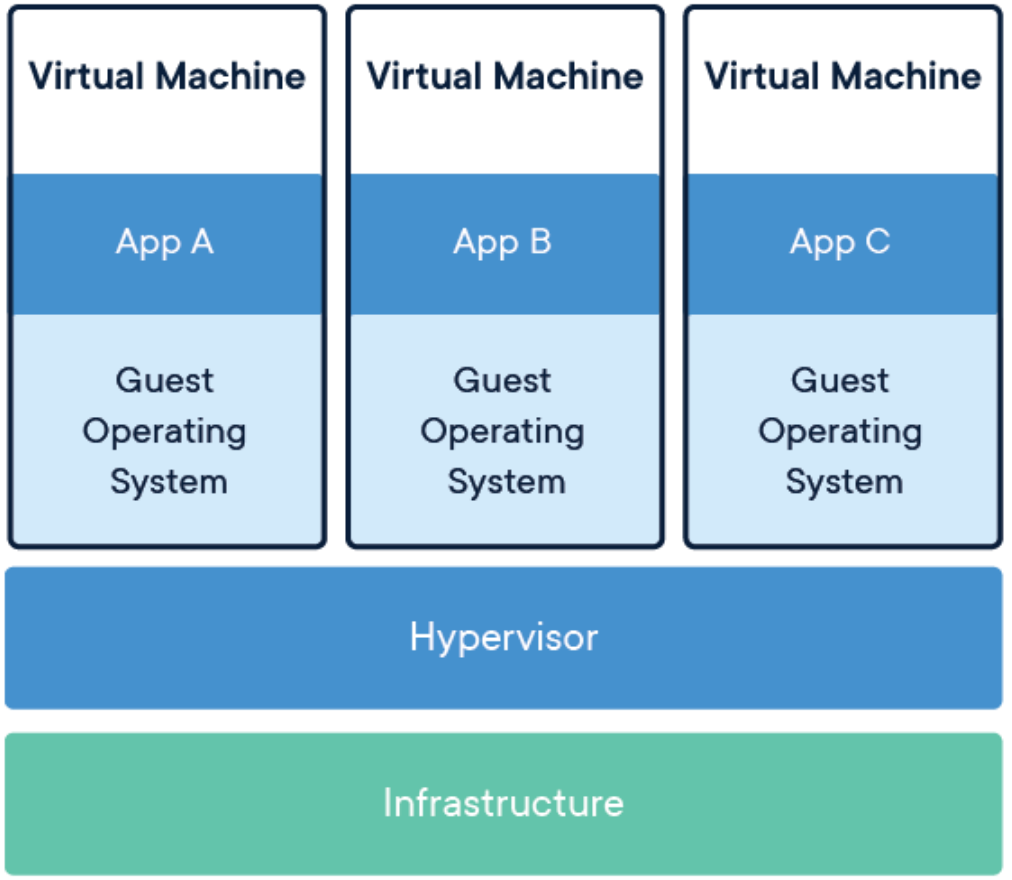
Agenda

1. Theory part
   - What is docker and containerization
   - Docker installation
   - Basic commands
   - Creating containers in theory

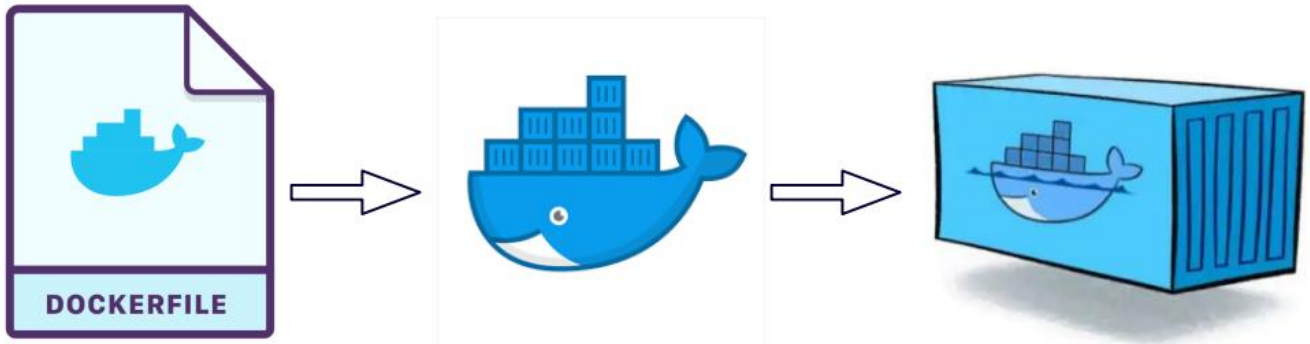
2. Practical part
   - Work with a single container
   - Application with several containers (docker-compose)

What is docker?





How to create a container?



Installing docker on the machine

Docker documentation: https://docs.docker.com/engine/install/

Installation on linux:

- `sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin`
- `sudo yum install -y docker`

- Starting the docker service*:
- `sudo service docker start`
  - **Disabling the docker service:**
  - `sudo service docker stop`
  - **Details of a running docker service:**
  - `docker info`
  - *exact commands depend on host OS

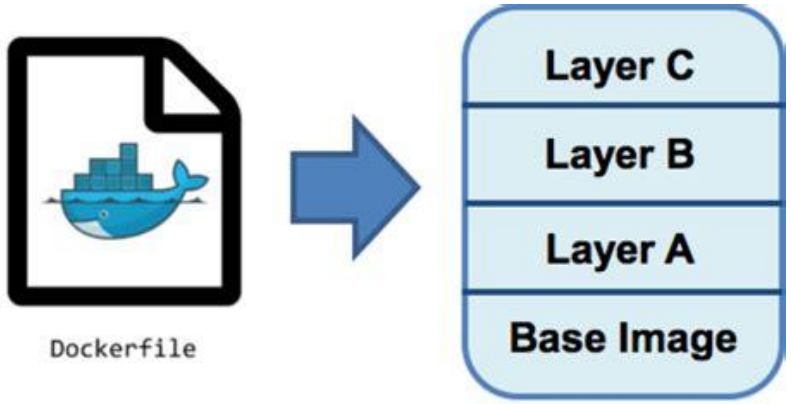
Basic commands

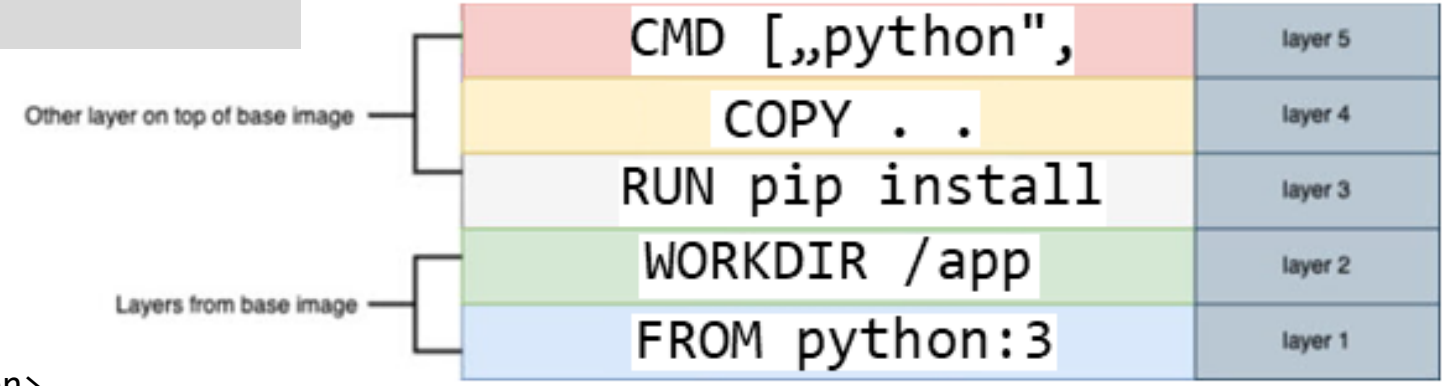
- `docker images` – list of all images
- `docker ps` – list of all running containers
- `docker ps -a` – list of all containers

- `docker run <image_name>:<tag> . - starts the container based on the selected image`
- `docker <command> --help – information about the executed command`
- `docker image rm <image_id> / docker rmi <image_id> - delete images`
- `docker rm <container_id> - remove container`
- `docker exec -it <container_id> <command> - connect to the console in the container`
- `docker start/stop <container_id> - enable/disable the container`

Dockerfiles and layers

```
FROM python:3
WORKDIR /app
RUN pip install --no-cache-dir --upgrade pip && \
    pip install --no-cache-dir flask
COPY . .
CMD [„python", „app.py"]
```





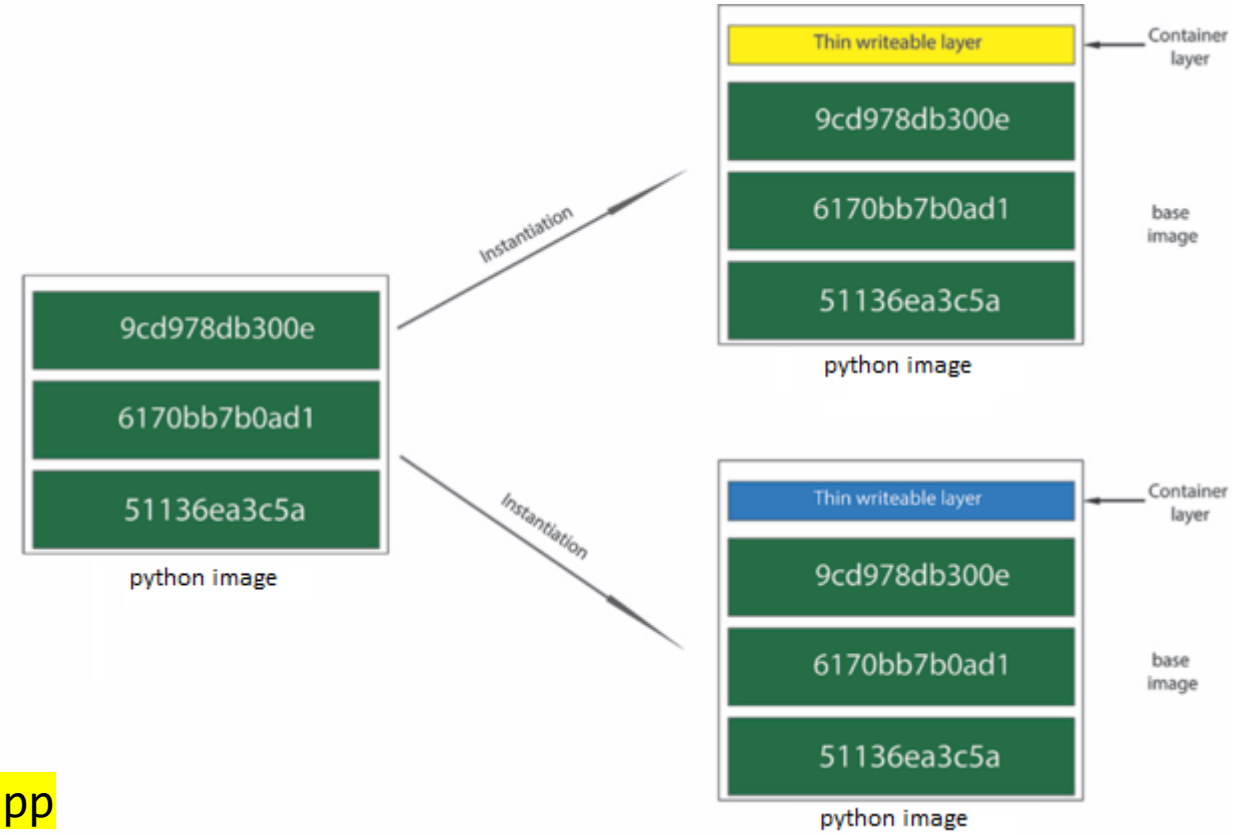
Create an image based on the Dockerfile:

```
docker build –t image_name:tag <dockerfile_location>
```

Dockerfile cheat sheet

New layer example

```
FROM python:3
WORKDIR /app
RUN pip install --no-cache-dir --
upgrade pip && \
    pip install --no-cache-dir
flask
COPY . /app
CMD [„python", „app.py"]
```



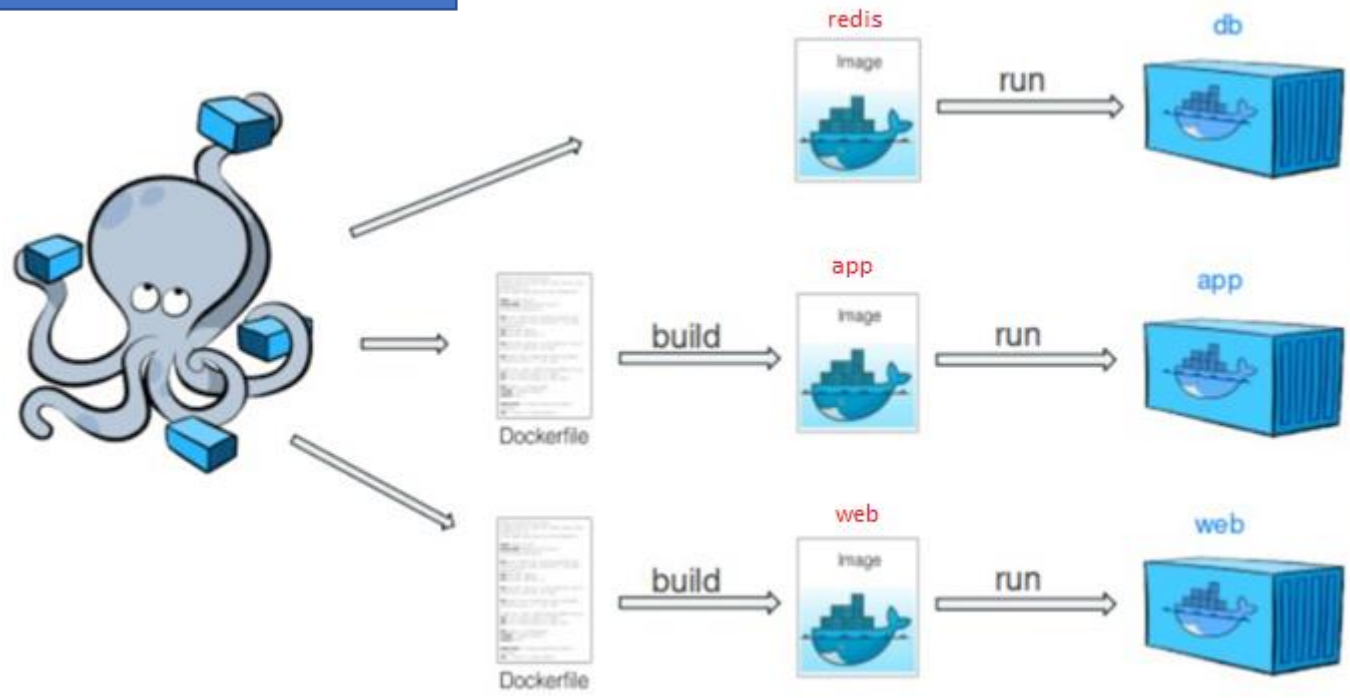

```
docker run -p 5000:5000 –e REGION=EU flask_app
```

```
docker run -p 5002:5000 –e REGION=NONEU flask_app_NON
```

Docker Compose

compose.yaml

```
version: "3.9"
services:
  web:
    build: .
    ports:
      - "8000:5000"
    volumes:
      - .:/code
    environment:
      FLASK_DEBUG: True
  redis:
    image: "redis:alpine"
```




```
docker compose up [-d] [-f <file_name>]
docker compose stop
docker compose down --volumes
```